Enumerating Potential Maximal Cliques via SAT and ASP

Tuukka Korhonen, Jeremias Berg, and Matti Järvisalo

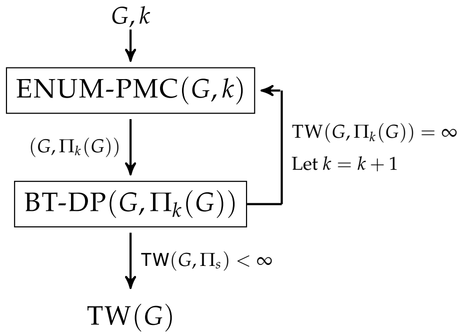
UNIVERSITY OF HELSINKI
FACULTY OF SCIENCE

Background

- Minimal triangulations define graph decompositions: tree decomposition, junction tree, phylogenetic tree, hypertree decompositions.
- BT algorithm of Bouchitté and Todinca: the best algorithm for computing optimal minimal triangulations.
- Bottleneck of BT runtime: enumeration of potential maximal cliques (PMCs).

Contributions

- Declarative encodings for enumerating PMCs with SAT and ASP solvers.
  - Three variants
- Experiments: treewidth and generalized hypertreewidth computation.
- Improvement over the original PMC enumeration algorithm.



Potential Maximal Cliques

Application: BT algorithm computes an optimal triangulation given the set of PMCs.

Definition 1: A set of vertices is a PMC if it is a maximal clique in some minimal triangulation.

Definition 2: Set of vertices $K \subset V$ is PMC if:

1. Any two vertices $u, v \in K$ are connected outside of K.
2. No vertex in $V \setminus K$ is connected to all vertices in K outside of K.

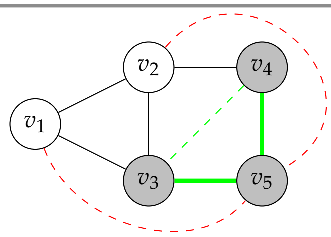
Encoding: Boolean variables P_i and C_{ij}

- $P_i = 1$ iff the vertex v_i is in PMC.
- $C_{ij} = 1$ iff the vertices v_i and v_j are connected outside of the PMC.

Encoding: Two conditions of PMC

1. $(P_i \wedge P_j) \rightarrow C_{ij}$
2. $\neg P_i \rightarrow \bigvee_{j=1}^{n} (P_j \wedge \neg C_{ij})$

Hard part:
Encoding the semantics of C_{ij}.



Connectedness: Let K be a set of vertices. Two vertices u, v are *connected outside of* K if there is a path $u, w_1, w_2, \ldots, w_k, v$ with no *intermediate* vertex w_i in K.

Encoding connectivity

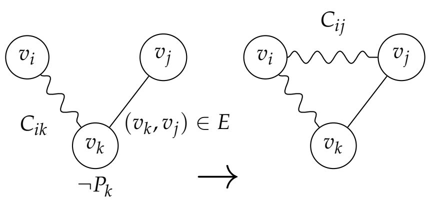
Both needed: *If* and *only if*.

If: $C_{ij} = 1$ if v_i and v_j are connected outside of the PMC.

Only if: $C_{ij} = 0$ if v_i and v_j are not connected outside of the PMC.

If direction: Propagate existing connectivity trough edges. $O(nm)$ additional clauses.

$$(C_{ik} \wedge (v_k, v_j) \in E \wedge \neg P_k) \rightarrow C_{ij}$$



Only if direction: Three variants

1. Path-length encoding in SAT:

- Additional variables $C_{ij}[k]$: v_i and v_j are connected outside of the PMC with a path of length k.
- Existence of shorter path required for existence of longer path.
- $O(n^3)$ additional variables.

2. ASP semantics:

- Existence of a path must have external support.
- No additional variables or constraints.

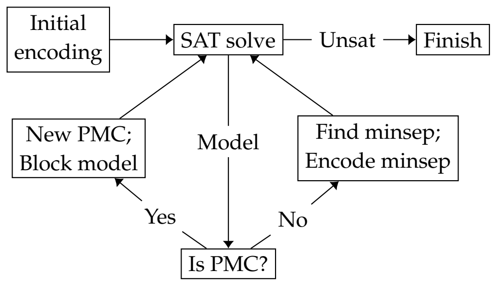
3. Lazy SAT:

- Use minimal separators to prove not connectedness when needed.
- Number of additional variables and clauses depends on the structure.

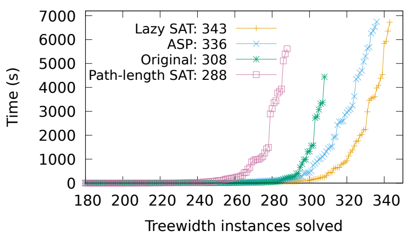
Lazy SAT does not include any constraints for the only if direction at first.

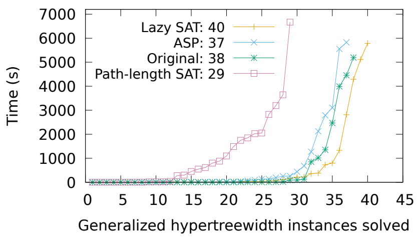
If a false PMC is found:

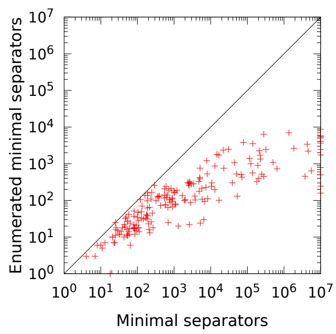
- $C_{ij} = 1$ for some not connected v_i, v_j.
- A *minimal separator* inside the false PMC proves that v_i and v_j are not connected.
- Find such minimal separator S and add encoding for it:
  - $(\bigwedge_{v_i \in S} P_i) \rightarrow M_S$
  - $\bigwedge_{i,j:\, S \text{ separates } v_i, v_j} M_S \rightarrow \neg C_{ij}$



Results







- Lazy SAT does not enumerate most of the minimal separators.
- Original PMC enumeration algorithm enumerates all minimal separators.

Runtime on *melon graphs* with $3n + 2$ vertices.

n	Original (s)	Lazy SAT (s)
6	0.46	0.04
7	3.27	0.06
8	26.80	0.06
9	261.20	0.09
10	2576.73	0.09
20	TO	0.38
50	TO	3.52
100	TO	39.00
200	TO	384.95

- Lazy SAT avoids the worst case complexity in melon graphs which have exponential number of minimal separators.